Università degli Studi di Milano

Facolta di Scienze e Tecnologie

*Corso di Laurea Magistrale in Sicurezza Informatica*

**

Time-Lock Encryption: una soluzione basata sulla crittografia ellittica e blockchain

**Relatore:** Prof. Stelvio Cimato

Tesi di:

Francesco Mancuso

Matricola: 942409

Anno Accademico 2020 – 2021

# Riassunto

In questo lavoro di tesi ho svolto un’analisi riguardante particolari sistemi di cifratura, denominati Time-Lock Encryption system. Questi sistemi sono stati teorizzati e sviluppati già diversi anni fa e portati avanti fino a giorni nostri. E’ stato quindi mio preciso obiettivo effettuare una panoramica delle tecniche esistenti, analizzare il loro funzionamento ed evidenziarne pregi e difetti. Soprattutto, è stata di fondamentale importanza verificarne l’effettiva efficacia, in relazione alle proprietà della time-lock encryption e alla situazione computazionale odierna. Tra tutti i modelli presentati, ne è stato proposto uno in grado di garantire le suddette caratteristiche. Si tratta di un sistema capace di rendere una certa informazione inaccessibile fino al raggiungimento di una deadline specificata. Di questa nuova proposta sono stati analizzati i principali punti di forza, sviluppata un’implementazione originale, valutata la sua reale efficacia e offerte eventuali modifiche e miglioramenti atti a risolverne le debolezze. Tutte le tecniche sviluppate e analizzate in questo lavoro di tesi sono state classificate in due macro categorie: agenti e puzzle. Nella prima parte del mio elaborato ho svolto una panoramica dei principali sistemi che fanno ampio uso sia di agenti sia di puzzle. Dall’analisi è emerso che quasi nessuno di questi sistemi è tutt’oggi in grado di garantire una cifratura sicura a tempo. Questo perché le capacità computazionali odierne sono molto più importanti di quanto lo fossero in passato. La tecnica cardine di questo lavoro di tesi unisce l’uso delle curve ellittiche con cifrario asimmetrico ElGamal con le caratteristiche intriseche della blockchain Ethereum. Ho effettuato uno studio della proposta originale fornendo un’implementazione del protocollo, riadattando e ampliando i concetti principali esposti. Questo sistema lavora in un ambiente decentralizzato dove un insieme di partecipanti competono per risolvere un certo problema computazionale: trovare la chiave privata di una curva ellittica partendo da una chiave pubblica. La sfida sta nel identificare una adeguata curva ellittica, il cui tempo di rottura debba essere il più vicino possibile al tempo desiderato. Gli aspetti chiave di questa implementazione sono la funzione di Icart, lo Smart Contract e l’algoritmo di Pollard’s Rho. La funzione di Icart è fondamentale per mappare un valore in un punto della curva, lo Smart Contract è l’unico elemento che permette la condivisione di informazioni tra i partecipanti e la risoluzione del problema computazionale, mentre l’algoritmo di Pollard’s Rho rimane il principale e più efficiente metodo in grando di stimare la debolezza di una curva ellittica prestabilita. Lo sviluppatore, tramite contratto, rende disponibile per tutti una chiave pubblica, mentre i partecipanti, richiamando apposite funzioni, cercheranno la corrispondente chiave privata, grazie alla quale sarà possibile decifrare il messaggio segreto. Nel caso in cui la chiave privata fosse corretta, riceveranno un reward adeguato. Questo sistema è particolarmente dipendente dalle capacità computazionali dei computer che lo eseguono, il che rappresenta anche una sua grande debolezza. Infatti, il mittente del messaggio deve essere in grado di effettuare una previsione delle risorse degli avversari. Il problema dell’algoritmo di Pollard’s Rho riesiede nel fatto che effettua solo una stima del tempo di rottura della curva perché influenzato da fattori casuali che ne rendono variabili i risultati. Per evitare che qualche partecipante “privilegiato” possa essere favorito nella ricerca della chiave ho pensato di applicare il principio alla base del Proof-of-Work dei sistemi su Blockchain, obbligando tutti i membri della rete ad attendere il tempo richiesto per la decifratura, indipendentemente dal loro potere computazionale. L’idea è sfruttare il Proof-of-Work usando come base un seme presente nello Smart Contract. Quindi, oltre a fornire la chiave privata candidata, il partecipante dovrà trovare un certo numero di nounce (partendo da un seme fornito dallo sviluppatore dello Smart Contract), tale che l’hash calcolato sull’attuale nounce, concatenato a quello precedente, sia minore di un target fissato (target scelto per rispettare il tempo stabilito). Esattamente come funziona per la rete Bitcoin. Una volta che un partecipante ha trovato tutti i valori richiesti, li invia allo Smart Contract e, nel caso fossero corretti, riceverà un reward adeguato. La chiave pubblica verrà ottenuta applicando la funzione di Icart all’ultimo nounce e verrà usata per scoprire la relativa chiave privata per decifrare il messaggio finale segreto. In questo modo, il tempo richiesto dal processo di time-lock encryption corrisponderebbe a:

dove è il tempo richiesto per ottenere tutti i nounce, mentre è il tempo di rottura della curva prescelta.

Ci sono due principali problemi in questa soluzione:

* Tutto il protocollo di time-lock encryption diventerebbe dipendente, non solo dalla curva prescelta, ma soprattutto dalla scoperta dei nounce. In altri termini, lo sviluppatore dello Smart Contract potrebbe scegliere una curva con tempo di rottura bassissimo e obbligare tutti i partecipanti ad attendere la deadline desiderata sfruttando solo il Proof-of-Work.
* Un partecipante, non possedendo capacità computazionali elevate, potrebbe sentirsi scoraggiato nel dover trovare tutti valori richiesti. Una possibile soluzione è quella di slegare i nounce dalla curva, ovvero un partecipante più debole potrebbe aspettare che qualcun altro risolva il Proof-of-Work (ricevendo in cambio un reward) e, quando la chiave pubblica viene resa disponibile, mettersi in gioco cercando di rompere la curva, ottenendo, in caso di esito positivo, un altro reward.

Al momento queste soluzioni sono un piccolo punto di partenza da cui poter sviluppare una promettente implementazione in grado di garantire le caratteristiche della time-lock encryption.

Struttura della tesi

* Nel Capitolo 1 viene introdotto l’argomento “time-lock encryption”, illustrando i concetti preliminari, le possibili applicazioni e l’attuale stato dell’arte. Vengono poi definiti quelli che sono i risultati attesi di questo lavoro di tesi.
* Nel Capitolo 2 viene affrontata una prima modalità di time-lock encryption basata sull’utilizzo, prima di agent non fidati, poi di trusted agents. Verranno analizzate nel dettaglio tecniche come il Secret Sharing, Key Sharing (decentralizzate con partecipanti non fidati) e tecniche come la Time–Lapse Cryptography e il Conditional Oblivious Transfer (centralizzate con partecipanti fidati ma curiosi).
* Nel Capitolo 3 viene affrontata una seconda modalità di time-lock encryption basata sulla risoluzione di puzzle complessi che richiedono un determinato quantativo di tempo prestabilito. Verranno analizzati nel dettaglio problemi computazioni basati su quadrati ripetuti e sul cifrario RC5 con chiave modulabile.
* Nel Capitolo 4 viene analizzato lo schema di time-lock encryption in ambiente Bitcoin basato sul Witness Encryption. Dopo la spiegazione del suo funzionamento dall’articolo originale, verrà approfondita l’implementazione proposta nell’articolo successivo e analizzati i problemi che ne conseguono.
* Nel Capitolo 5 verrà analizzato lo schema di time-lock encryption basato su curve ellittiche e Blockchain Ethereum. Verrà fornita un’implementazione di uno Smart Contract scritto nel linguaggio Solidity, un’approfondimento della funzione di Icart per la mappatura di un messaggio in un punto di una curva e un’implementazione del Pollard Rho Attack scritto in Java per la valutazione delle vulnerabilità di una curva.
* Nel Capitolo 6 verrà svolta un’analisi sull’implementazione sviluppata riguardante la sua reale efficacia. Sarà inoltre mio interesse proporre delle modifiche all’implementazione per migliorare i problemi riscontrati dall’analisi stessa.
* Nel Capitolo 7 verrà fatto un riepilogo di tutto il lavoro svolto e verrà svolto un riassunto conclusivo di tutte le tecniche analizzate nel presente lavoro.